2010

Simulating Human Activities for Synthetic Inputs to Sensor Systems

Jan Allbeck
University of Pennsylvania

Norman I. Badler
University of Pennsylvania, badler@seas.upenn.edu

Allbeck, J. & Badler, N., Simulating Human Activities for Synthetic Inputs to Sensor Systems, *Distributed Video Sensor Networks*, Springer, 2010, doi: 10.1007/978-0-85729-127-1_13

Simulating Human Activities for Synthetic Inputs to Sensor Systems

Abstract
We are developing human activity simulations that could be used to test distributed video sensor networks. Our ultimate goals are to build statistical models of pedestrian density and flows at a number of urban locations and to correlate those flows with population movement and density models represented in a spatiotemporal modeling system. In order to create known populace flows, we have built a virtual populace simulation system, called CAROSA, which permits the authoring of functional crowds of people going about role-, context-, and schedule-dependent activities. The capabilities and authoring tools for these functional crowd simulations are described with the intention of readily creating ground truth data for distributed sensor system design and evaluation.

Disciplines
Computer Sciences

Comments
Allbeck, J. & Badler, N., Simulating Human Activities for Synthetic Inputs to Sensor Systems, *Distributed Video Sensor Networks*, Springer, 2010, doi: 10.1007/978-0-85729-127-1_13

Chapter 13
Simulating Human Activities for Synthetic Inputs to Sensor Systems

Jan M. Allbeck and Norman I. Badler

Abstract We are developing human activity simulations that could be used to test distributed video sensor networks. Our ultimate goals are to build statistical models of pedestrian density and flows at a number of urban locations and to correlate those flows with population movement and density models represented in a spatiotemporal modeling system. In order to create known populace flows, we have built a virtual populace simulation system, called CAROSA, which permits the authoring of functional crowds of people going about role-, context-, and schedule-dependent activities. The capabilities and authoring tools for these functional crowd simulations are described with the intention of readily creating ground truth data for distributed sensor system design and evaluation.

Keywords Crowd simulation · Virtual agents · Human activity · Computer animation

1 Overview

Current urban populace modeling lacks appropriate computational methods to realistically represent complex dynamic interactions between the built environment, cultural artifacts, and population behavior. Our objective is *to develop new methods to rapidly create a model of pedestrian flows correlated with appropriate spatiotemporal environment models*. Such models admit both virtual simulation and

J.M. Allbeck (✉)
George Mason University, 4400 University Dr., MSN 4A5, Fairfax, VA 22030, USA
e-mail: jallbeck@gmu.edu

N.I. Badler
University of Pennsylvania, 3300 Walnut St., Philadelphia, PA 19104-6389, USA
e-mail: badler@seas.upenn.edu

B. Bhanu et al. (eds.), *Distributed Video Sensor Networks*,
DOI 10.1007/978-0-85729-127-1_13,

understanding of a real world urban setting wherein individuals engage in culturally realistic and plausible behaviors and movements. These models also admit anomaly detection where pedestrian flows (or lack thereof) would be perceived as unusual or otherwise significant occurrences, perhaps alerting and directing the attention of human operators or other automated observers.

Our project entails the creation of a computer software system for modeling, authoring, and controlling an urban populace. Our approach is synthetic in that we are developing computational simulation tools, but it is also experimental because we can use the simulation to model known conditions and validate its predictions. It is also general in that the developed methodology and system can be applied to a variety of urban terrain environments and human terrain conditions.

This chapter is structured as follows. In the next section we will discuss some related crowd simulation work. We will then give an overview of the CAROSA (Crowds with Aleatoric, Reactive, Opportunistic, and Scheduled Actions) populace simulation framework, followed by some key components of the framework and how they enable easily authored situated, contextual populations. We conclude with future directions for this work in the context of distributed sensor systems.

2 The CAROSA System

Distributed video sensing of human activities in an urban setting requires a fundamental understanding of what human behaviors are likely, normal, or anomalous in such an environment. Since it is difficult and costly to run actual actor-based situations (e.g., as portrayed in the movie *The Truman Show*), computer simulations can often provide the only realistic route for synthetic urban populace stimuli. Large-scale simulations are frequently done macroscopically, with limited models and graphics of the urban inhabitants and the architectural context, and thus fail to give a personal view of urban life, which is crucial for appropriate sensing of and responses to observed behaviors. While many computer graphics research groups are producing animated crowds of hundreds and even thousands of people, often the agents wander (apparently aimlessly) over the traversable areas.

Urban environments are complex dynamic places, populated by individuals with roles, goals, schedules, and diverse but not entirely random behaviors that are driven by many forces: social, spatial, personal, and immediate needs. Urban movements are semi-structured and rather mundane, and lack the time-critical focus of panic and evacuation. People work, play, visit, shop and loiter. We have developed the CAROSA system [20] to cover a broad range of relevant issues related to this basic axiom, concentrating on both low level individual motions and high level selection and control of individual behaviors predicated on scheduled, reactive, opportunistic and stochastic actions. Actions are tied to semantic data associated with environments (e.g., buildings or rooms), thus providing flexibility for agents to function without fixing them to spatial locations or animation scripts [27]. Human schedules interact with spaces, work and other cycles [19]; computer graphics simulations mostly lack these socio-cultural and functional aspects of human behavior.

The creation of these heterogeneous populations with contextual behaviors needs to be feasible and, even better, based on empirical evidence. We do not want to demand that scenario authors be expert programmers. Likewise, defining these populations should not be an arduous, never-ending process requiring levels of detail beyond what is immediately important to the scenario author. Especially in a distributed sensor simulation application, spatiotemporally *structured randomness* may be a distinguishing characteristic required of a crowd synthesizer.

3 Related Work

Many movies and games use software such as Massive™ to build background characters [15]. Massive™ provides mechanisms for creating and executing rules that govern the character behaviors that are replicated for large groups. While creating and refining these rules still takes time and skill, the software makes construction of relatively homogeneous crowds (with some statistical variations) much easier. While this is well suited to battle scenes where interactions are limited, scenes that require functional, contextual characters are not feasible.

In many games and simulations, populations of non-player characters (NPCs) are programmed to follow a path and perhaps perform a few behaviors. They might also react to a certain set of stimuli, but they do not generally interact with objects in the environment. They lack context, often only existing when in the player's field of view [3].

Crowd simulations based on Helbing's empirical Social Forces model [9] use repulsion and tangential forces to simulate interactions between people and obstacles, pushing behaviors, and flow rates. More realistic human movement in low- and medium-density crowds can be obtained using rule-based models [24]. Cellular-automata models [6] are fast and simple to implement, but disallow contact between agents by explicitly prohibiting it. McDonnell et al. study the perception of character appearance and motion variations to enable the portrayal of more visibly interesting, reasonable populations [16]. Lerner et al. use real world data to fit behaviors into pedestrian simulations [13]. While these approaches have achieved visually interesting populations, the characters still lack larger purpose and context.

Other researchers have created variations in behavior contextually by defining regions in the environment where certain behaviors would be displayed [7, 12, 26]. For example, a theater is labeled with sit and watch behaviors. However, within these regions behaviors are uniform. Composite agents, on the other hand, use agent proxies to differentiate behaviors and reactions based on factors such as character priority or authority [31]. Others have also explored using social or psychological factors to vary behaviors; e.g., Thalmann et al. included two states for the characters, SOCIAL and HUNGRY [28]. Pelechano and Badler's work included leaders and followers as well as trained and untrained agents, but behavioral variants were limited to way-finding [22]. Both reactive and deliberative actions have been implemented [25] and decision networks have been used for action selection [32], though they require crafting the prior probabilities of each action in context.

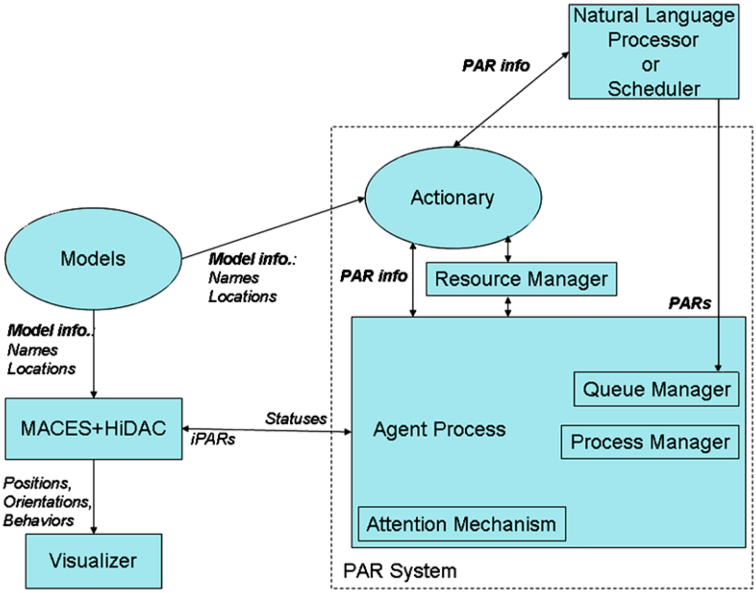

Fig. 1 The CAROSA framework

Our CAROSA framework includes many components common to simulators (see Fig. 1). Graphics models are displayed using OpenGL, Cal3D [5], or Ogre [18]. We are also using HiDAC+MACES as an underlying crowd simulator [21]. HiDAC+MACES navigates characters from one location to another while avoiding collisions and provides messages when objects and other agents are perceived. Calls to playback animation clips also filter through HiDAC+MACES to the graphics figures.

CAROSA's unique combination of components include an action and object repository called the Actionary, Agent Processes, a Resource Manager, and a Scheduler. Scenario authors can use the Scheduler to schedule actions found in the Actionary and link them to locations or specific objects. These actions are also associated with the agents that are to perform them. Agent Processes receive and process these actions, using the Resource Manager to allocate or bind any object parameters needed by the actions. The next few sections will describe these components and how they facilitate the creation of functional, heterogeneous populations.

4 Parameterized Representations

CAROSA uses the PAR (Parameterized Action Representation) system [2] for both parameterized actions and objects stored in a persistent hierarchic database: the *Actionary*. The Actionary currently contains more than 60 actions and nearly 100 object types. PARs use *preparatory specifications* that provide simple backward chaining, which greatly reduces the scenario-authoring burden. For example, most instantiated actions are tied to an object participant (e.g., sitting in a chair, using a computer, eating food, drinking coffee, etc.), and these actions have a preparatory specification that makes the agent walk to the associated object before the action can be performed.

General action definitions reference object class types as object participant parameters; e.g., a general definition of Sit includes Person-Supporting-Furniture. Similar to Smart Objects [11], PAR objects also include parameters that further specify how agents interact with them: e.g., where a character should stand to operate the object or how to grasp it. Functional characters can thus behave appropriately in context.

CAROSA extends the PAR representations to include four action types: scheduled, reactive, opportunistic and aleatoric. Scheduled activities arise from specified roles for individuals or groups and give agents purpose and a storyline. For example, the scenario author may schedule a meeting between two characters to show that they have an association. Purely scheduled actions alone would result in simulations that are too structured or robotic.

Reactive actions are triggered by contextual events or environmental constraints. Many of these behaviors arise from the underlying crowd simulator: e.g., an agent altering its heading or slowing to avoid collisions. Reactive actions, such as acknowledging someone as they pass by, are not handled by the crowd simulator. These reactions are specified and recognized in a rule-based Attention Mechanism. Reactive actions help to add life to simulations. Perceiving and interacting with a dynamic environment indicates that the characters are not focused solely on achieving a specific goal; they are also aware of the world around them.

Opportunistic actions arise from explicit goals and priorities. These need-fulfilling behaviors are akin to the hill-climbing behavior choices of characters in the popular game *The Sims* [29]. While our opportunistic actions are similar, the implementation is different. In *The Sims* current proximity to resources is heavily weighted when choosing an action, and time is ignored. We use both time and future proximities. For example, a character may be working in his office and have a non-emergent energy need and a meeting to attend in a few minutes. The character could then attempt to address the need by stopping by the lunch room for a cup of coffee on the way to the meeting. This requires them to leave a couple of minutes early and to know that the lunch room has proximity to the path to the meeting room. Finding an object resource is essentially done through a depth-first search where the depth limit is based on the need level. As the need level increases, so does the distance the agent is willing to go out of their way to fulfill the need. Like scheduled and reactive actions, opportunistic actions lead to the perception of functional characters.

Aleatoric actions are random but structured by choices, distributions, or parametric variations. The aleatoric quality arises from frequency of occurrence of subactions. We could use, e.g., the Bureau of Labor Statistics (BLS) American Time Use Survey, which encodes time distributions of many home, work, and leisure activities. Aleatoric behaviors are designed to provide reasonable variations in behavior without overt scheduling. Working in an office might include using a computer, speaking on a telephone, and filing papers. For many scenarios the exact timing of each sub-action is not important. The overarching *WorkInOffice* action should just look plausible.

5 Resource Management

Creating functional populations involves associating actions with related objects. Making these associations by hand could overwhelm a simulation author. We implemented a Resource Manager that automatically allocates objects. Resources are initialized at the onset of the simulation where the building[1] is defined. Object locations are set as the room that they are placed in and likewise the objects are listed as contents in those room objects. These relationships are then used to initiate resource management.

There are three different methods for allocating resources from a resource group. One method just allocates any available resource from the resource group to the agent. Another method specifies an object to be allocated. The last method specifies a preference function to determine the best object to allocate to the agent. Objects are not allocated to agents until both are in the same location or room. This leads to natural behaviors including failures: an agent must enter a room to know that there are insufficient chairs.

6 Roles and Groups

People's functions or purposes through the course of a day are highly correlated to their roles: *students* attend classes, *professors* work in their offices and lecture, *housekeepers* clean, *researchers* research, etc. Creating roles for simulated characters provides them focus and purpose and also creates simulations of heterogeneous populations. Defining groups of characters with common roles means reduced work for simulation authors.

When defining a role one can specify objects that the role should possess: e.g., a professor should have an office. An author does not need to specify an instance of an object, though they are permitted; it is sufficient to specify a class like *office*. Possessions are transitive: if you have an office, you also have the objects in the office. When a role has a possession specified, every character with that role is allocated (by the *Resource Manager*) a possession of that type during the initialization of the simulation. Whenever a character initiates the performance of her default behavior, she first checks to see if an object of the type needed is located in her possessions. If it is, she uses that object. If it is not, the next method is tried.

[1] Work is in progress to extend this scheme to large exterior environments; the modeling principles remain the same.

This method also uses the association of object types with default actions. For example, researchers in our scenario need laboratory desks in order to do their research. As we did with professors, we could say that all researchers should possess a lab desk, but in a university setting where resources are limited it might be more likely for researchers to take whatever desk is available. We do this by indicating that the action *research* requires an object participant of the type *LabDesk*. When a researcher initiates this default action, an object of type *LabDesk* is allocated to her. If no object can be allocated, a failure is produced and the action will not be performed. This might lead to the character performing a different action or just standing still (or idling as a default action) until an appropriate object can be allocated.

A key aspect of CAROSA is facilitating the creation of heterogeneous populations. To achieve this, we need to provide a scenario author a way to define groups as well as individuals. An author can name a group and provide the number of members in it, or an author can create individual characters and assign them to groups. In both cases, the information is stored in the Actionary. As needed, agents are created to fulfill the indicated membership numbers.

If the name of a group happens to correspond to the name of a role, then all of the members of the group are automatically assigned that role and inherit all of the default behaviors and specifications for it. There can be more than one group per role and there may be groups that do not indicate a role. An author may want to do this for a couple of reasons. First, groups can be assigned actions, so there may be a group of students that have class at the same time and can be scheduled as a group. There might also be meetings that group members of various roles attend, such as project meetings. When a group action is specified, it is simply copied and placed on the action queue of each member in the group. Normally, a location for the action is given as well as the type of object participant. The Resource Manager then allocates the needed object resources as each participant arrives at the location. Similar to real world email lists, creating groups that correspond to meaningful collections of characters enables group events to be quickly and easily scheduled. Naturally, scheduling conflicts can and do occur just as in the real world. The CAROSA framework resolves these conflicts seamlessly based on the actions' priorities as specified in the PARs. If the actions have the same priority, the first one on the queue will be performed.

Another reason for creating more than one group per role and groups that do not correspond to roles is plausible group reactions. Reactive actions can both be assigned to groups and groups can be the stimulus of reactive actions.

We are not the first to consider roles definitions for virtual characters [10, 23]. Our efforts, however, focus less on the social interactions of the characters. For our work, roles are vital to facilitating the creation of functional, heterogeneous populations. Likewise, groups have been a part of virtual character research [4, 12, 17, 28]. For the most part, these previous efforts have focused on collision detection and dynamic group behaviors. In the CAROSA framework, groups generally have common purposes or functions. They may or may not gather during a simulation. For example, a simulation may contain a group of housekeepers or maintenance workers to clean and maintain the building, but these characters may never gather into a visual group.

7 Scenario Authoring

Beyond creating functional, heterogeneous animated crowds, our goal for the CAROSA framework was to facilitate scenario authoring by non-programmers. Certainly to simulate a virtual environment, one will still require models and animations, but once a repository of these items has been created and a populated Actionary obtained and linked to them, a Subject Matter Expert (SME) should be able to use these building blocks to construct a simulation.

When scheduling actions, a SME only needs to specify the PAR action that should be performed, which character or group of characters should perform it, what objects or locations might participate in the action, and when it should be performed. In fact, the actions can either be simple PARs or complex lists of PARs composed together. A specific object participant can be specified, such as sit in *Chair_4* or a location can be given, such as sit in *ClassRoom_1*. If a location is specified, a chair instance will be allocated by the Resource Manager when the character arrives in the specified location. If the required resource cannot be allocated, a failure is reported to the Agent Process and the action is removed from the queue. In the future, a more powerful planner could be used to attempt to acquire the resource needed.

We are currently using the calendars of Microsoft Outlook® as a scheduling interface. Calendars can be created for groups of characters or individuals. Activities, locations, and times are then simply entered on the calendars. The GeniusConnect [8] software package then links these calendars to tables in the *Actionary* MySQL database. Activities are read from the database and assigned to the appropriate characters by placing them on the appropriate *Action Queues* of the *Agent Processes* (See Fig. 1).

We are also using the interfaces of Microsoft Outlook® to enable SME's to create groups and define roles. These definitions are created through Microsoft Outlook®'s Contacts and Tasks panels. We have defined a Quantity field within the Contacts panel that can be used to specify the number of members of each group. Required possessions can be entered in the Office field. Default behaviors are associated with roles through the Tasks panel, simply by entering the action name in the Subject field and the role in the Contacts field. We could also create macros and other interface elements to streamline the process.

We have constructed simple custom Graphical User Interfaces (GUI's) for authoring reactive, opportunistic, and aleatoric actions. These GUI's are directly connected to the Actionary. The drop-down lists are populated from table entries in the database and submitting new actions of these types writes directly to database tables. These newly created actions can then also be referenced from Microsoft Outlook®, e.g., to schedule aleatoric actions.

The PAR representation of objects and actions provide semantic foundations that are referenced when authoring scenarios through these interfaces. The Resource

Manager provides a means of filling in information and tying the simulation to the environment. This allows a SME to concentrate on aspects of the simulation that are directly relevant to a scenario including the heterogeneity and functionality of the population.

8 Example Simulation

As an initial test-bed for the CAROSA framework, we simulated a university building environment. The environment is based on one floor of an engineering building at a university. It includes laboratories, classrooms, hallways, offices, a restroom, and a lounge (See Fig. 2). The occupants of the building include professors, students, researchers, administrators, housekeepers, and maintenance personnel. Actions in this environment include working in an office, lecturing, attending class, lounging, cleaning, inspecting, researching, waving, eating, drinking, going to the restroom, picking up objects, as well as others. There is also collision-free locomotion.

Characters in the simulation adhere to their schedules as created by the scenario author through the Scheduler, but they also greet each other as they pass by and attend to their needs through opportunistic actions. If a portion of their day is unscheduled, the characters revert to their default behaviors, which in many cases are aleatoric actions that provide ongoing reasonable behavior.

We ran many simulations of this environment and noted several emergent behaviors. For example, students tended to gather in the same areas because of the resources available and therefore tended to walk to classes together. Furthermore when groups were instructed to react to other group members by waving, students would greet each other before a class, which seems like reasonable behavior. Students also tended to go directly for food and coffee after classes. Because need levels

Fig. 2 Sample scenario based on activity in a university environment

are currently all checked together, needs tend to be fulfilled at the same time. For example, a character might go to the restroom and then to get coffee. Again this emergent behavior seems quite plausible.

9 CAROSA Summary

The CAROSA framework facilitates the creation and simulation of functional, heterogeneous populations. Characters in our simulations do more than navigate from one location to another without colliding with each other or obstacles: characters have roles that provide them purpose. Their behaviors are performed in proper context with the object participants. In fact, because actions reference PAR objects and not coordinates directly, the environmental setting of the simulation can be completely rearranged without requiring additional work from the simulation author. Objects can even be added or removed and the simulation will still run, although some actions may not be performed if there is a shortage of resources.

Through the use of the PAR representations, a resource manager, and definitions of roles and groups, we have created a framework in which actions can be scheduled in forms analogous to the scheduling of real people. To demonstrate this, we have connected the CAROSA framework to Microsoft Outlook®. Through this widely used interface for scheduling real human and group activities, a subject matter expert can author simulations. To add additional richness and reasonable behaviors to simulations, we have also created simple custom GUI's for authoring reactive, opportunistic, and aleatoric actions.

Additional development is still needed for the CAROSA framework. In particular, we would like to increase the scale of the populations. Our university simulation can run 30 characters in real-time on a standard PC. The most costly algorithm in the CAROSA framework is the calculation of opportunistic actions. Scheduling the fulfillment of a need can require searching through time for a gap in the character's schedule as well as through the environment to find a resource near a path the character will be traveling. We are considering caching these paths or locations or perhaps using stored waypoints that act as road signs for resources. We would also like to explore running each character of a simulation on a different processor with one additional processor used by the underlying HiDAC+MACES crowd simulator or another engine to handle collision detection. Fortunately in a simulated sensor input setting, real-time input is not as important a requirement as reasonable populace activities.

Further, we are currently working on tying the CAROSA framework to other underlying engines such as the game engines Ogre and Dark GDK. Such engines would provide us with better computer graphics and action animations. Ideally, we would like the underlying animation engine to include inverse kinematics for the characters, allowing the PAR representation to drive more detailed human/object interactions. This would require additional semantic labeling of the objects in the environment (i.e., creation of sites), but, as objects can be reused in many simulations, the cost would grow linearly or slower with the size of the urban area being modeled.

10 Input to Distributed Sensor Networks

Ultimately the input media for this project should be multiple real-time video data streams from a distributed suite of surveillance cameras, of both fixed and movable types. If fixed, their locations are known in advance and can be readily geo-located in a digital urban map. If movable, their present position could be sensed and reported via GPS and compass readings. CAROSA's use as a simulator for distributed sensor system input has the advantage that the environment may be readily compartmentalized (e.g., into rooms) or into separated buildings. By positioning fixed or dynamic synthetic cameras in various desired locales within the 3D space, each camera will perceive only the activities within its local view. Meanwhile CAROSA simulates the entire environment (globally) with structure and consistency in character activities across the various spaces. By synthesizing populace motions and behaviors, we can both control the behaviors in the overall model, and have known ground truth against which to compare and evaluate sensor output and analysis results.

Since we are still in the early stages of our work CAROSA allows us to bypass the important but difficult question of actually doing real-time video pedestrian tracking in a video stream in favor of the more global scale problem of building the requisite computational semantic structures in which such videos can be interpreted. In addition, obtaining unrehearsed, public, and urban surveillance video is a non-trivial undertaking. Using CAROSA, we can simulate a populace and base input features to be measured and descriptions produced (e.g., [1]) on flows rather than tracking individuals in crowds. The latter is a hard problem (e.g., [14, 30]), but we feel that this level of detail is not necessary for the accomplishment of our goals. In addition, an ability to track one person in a video sequence provides no explicit connection or correlation to any other people that may appear in other video streams; i.e., our interest is not in fusing tracks of individuals but rather of observing whether global and distributed pedestrian motions are consistent with spatiotemporal models of expected pedestrian flows.

11 Summary

The principal challenge in creating a CAROSA simulation and obtaining the corresponding spatiotemporal features is data collection and organization for people, places, and events. The data needed to seed this project should be public and local to our environment, so that we may both measure and predict the cost of scaling data collection to a different and larger urban area. We have access to, and can observe, e.g., typical student schedules, room locations, and work patterns at the University of Pennsylvania, so these could be a starting point for our databases. This process protects individual privacy while providing us with an external verifiable measure of simulation viability.

We have described a framework for building a baseline system that creates simulation-generated sensor data from a dynamic spatiotemporal model. We believe this baseline can be extended to use actual computer vision sensor data from real cameras. This framework could be a major step toward an ultimate goal of a real-time predictive model of urban human behaviors that can supply both intelligence and geospatial reasoning to military and civil needs.

Acknowledgements Partial support for this effort is gratefully acknowledged from the U.S. Army "SUBTLE" MURI, a Lockheed-Martin Corporation Strategic Technology Thread Grant, and George Mason University. We also appreciate donations from Autodesk and nVidia. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

1. Allbeck, J., Kipper, K., Adams, C., Schuler, W., Zoubanova, E., Badler, N., Palmer, M., Joshi, A.: ACUMEN: Amplifying control and understanding of multiple entities. In: Autonomous Agents and Multi-Agent Systems, pp. 191–198 (2002)
2. Bindiganavale, R.: Building parameterized action representations from observation. Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA (2000)
3. Brockington, M.: Level-of-detail AI for a large role-playing game. In: Rabin, S. (ed.) AI Game Programming Wisdom, pp. 419–425. Charles River Media, Inc., Hingham (2002)
4. Brogan, D., Hodgins, J.: Group behaviors for systems with significant dynamics. Auton. Robots **4**, 137–153
5. CAL3D. 3D Character Animation Library: http://home.gna.org/cal3d/, Last visited May 2009
6. Chenney, S.: Flow tiles. In: ACM SIGGRAPH/ Eurographics Proceedings of Symposium on Computer Animation, pp. 233–242 (2004)
7. DePaiva, D.C., Vieira, R., Musse, S.R.: Ontology-based crowd simulation for normal life situations. In: Proceedings of Computer Graphics International, Stony Brook, NY, 2005, pp. 221–226. IEEE, New York (2005)
8. GeniusConnect: GeniusConnect. http://www.geniusconnect.com/articles/Products/2/3/, Last visited May 2009
9. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature **407**, 487–490
10. Isbister, K., Hayes-Roth, B.: Social implications of using synthetic characters: an examination of a role-specific intelligent agent. Knowledge Systems Laboratory, Stanford University, Stanford, CA (1998)
11. Kallmann, M., Thalmann, D.: Direct 3D Interaction with smart objects. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, London, United Kingdom, 1999, pp. 124–130. ACM, New York (1999)
12. Lee, K.H., Choi, M.G., Hong, Q., Lee, J.: Group behavior from video: A data-driven approach to crowd simulation. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, 2007, pp. 109–118 (2007)
13. Lerner, A., Fitusi, E., Chrysanthou, Y., Cohen-Or, D.: Fitting behaviors to pedestrian simulations. In: Symposium on Computer Animation, New Orleans, LA, 2009, pp. 199–208. ACM, New York (2009)
14. Ma, Y., Cisar, P., Kembhavi, A.: Motion segmentation and activity representation in crowds. Int. J. Imaging Syst. Technol. **19**(2), 80–90
15. MASSIVE_SOFTWARE_Inc.: 3D animation system for crowd-related visual effects, http://www.massivesoftware.com, Last visited Oct. 2009
16. McDonnell, R., Micheal, L., Hernandez, B., Rudomin, I., O'Sullivan, C.: Eye-catching crowds: saliency based selective variation. In: ACM SIGGRAPH, New Orleans, Louisiana, 2009, pp. 1–10. ACM, New York (2009)

17. O'Sullivan, C., Cassell, J., Vilhjalmsson, H., Dobbyn, S., Peters, C., Leeson, W., Giang, T., Dingliana, J.: Crowd and group simulation with levels of detail for geometry, motion and behavior. In: Third Irish Workshop on Computer Graphics (2002)
18. Orge: http://www.ogre3d.org/, Last visited January 2010
19. Parkes, D., Thrift, N.: Times, Space, and Places: A Chronogeographic Perspective. Wiley, Chichester (1980)
20. Pelechano, N., Allbeck, J., Badler, N.: Virtual Crowds: Methods, Simulation, and Control. Morgan and Claypool Publishers, San Rafael (2008)
21. Pelechano, N., Allbeck, J.M., Badler, N.I.: Controlling individual agents in high-density crowd simulation. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA, San Diego, CA, 2007. ACM Press, New York (2007)
22. Pelechano, N., Badler, N.: Modeling crowd and trained leader behavior during building evacuation. IEEE Comput. Graph. Appl. **26**(6), 80–86
23. Prendinger, H., Ishizuka, M.: Social role awareness in animated agents. In: 5th International Conference on Autonomous Agents, pp. 270–277. ACM Press, New York (2001)
24. Reynolds, C.: Flocks, herds, and schools: A distributed behavior model. In: Proceedings of ACM SIGGRAPH, pp. 25–34 (1987)
25. Shao, W., Terzopoulos, D.: Autonomous pedestrians. In: Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (Los Angeles, California, 2005), pp. 19–28. ACM Press, New York (2005)
26. Sung, M., Gleicher, M., Chenney, S.: Scalable behaviors for crowd simulation. Comput. Graph. Forum **23**(3), 519–528
27. Thalmann, D., Musse, S.R.: Crowd Simulation. Springer, Berlin (2007)
28. Thalmann, D., Musse, S.R., Kallmann, M.: Virtual humans' behavior: individuals, groups, and crowds. In: Proceedings of Digital Media Futures, pp. 13–15 (1999)
29. Wright, W.: The Sims. Electronic Arts, 1st edn. (2000)
30. Ya-Dong, W., Jian-Kang, W., Ashraf, A.K., Wei-Min, H.: Tracking a variable number of human groups in video using probability hypothesis density. In: Proceedings of the 18th International Conference on Pattern Recognition, vol. 03. IEEE Computer Society, Los Alamitos (2006)
31. Yeh, H., Curtis, S., Patil, S., van den Berg, J., Manocha, D., Lin, M.: Composite agents. In: Proceedings of Symposium on Computer Animation, Dublin, Ireland, 2008. Eurographics Association, Zurich (2008)
32. Yu, Q., Terzopoulos, D.: A decision network framework for the behavioral animation of virtual humans. In: Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, California, 2007, pp. 119–128. Eurographics Association, Zurich (2007)